
48K
Spectrum
N.º 9
Monser
sinclair
AÑO 1
REVISTA
MENSUAL
COLECCIONABLE
P.V.P.
375
INDEXADOS EN TU SPECTRUM
ANALISIS SOFT
HERBERT'S
DUMMY RUN
VIII PARTE
CURSO CODIGO
MAQUINA
Frank N' Stein



Sumario:

AÑO I - N.º 9 - 1985

DIRECTOR
José Nieto Rubio

COORDINADOR
Félix Santamaría Avila

SUPERVISOR SOFTWARE
Gustavo Cano Muñoz

DISEÑO
Angélica Arce

REDACCION Y COLABORADORES
José Luis Roig
Victoria Aguilar
Javier González

PORTADA
Mauro Novoa

EDITA
MONSER, S. A.

DIRECTOR EDITORIAL
J. L. Cano Regidor

REDACCION, ADMINISTRACION Y PUBLICIDAD
Argos, 9
28037-MADRID
Tel. 742 72 12/96

PUBLICIDAD Y SUSCRIPTORES
Yolanda Bardillo

FOTOCOMPOSICION
CRISOL, S. A.
Virgen del Val, 48

FOTOMECANICA
IMAGEN

IMPRIME
GRAFICAS IBARRA

DISTRIBUCION
DISPRENSA
Eduardo Torroja, 9

Depósito legal: M-10.328-1985

Se solicitará control O.J.D.

NOTICIAS

OTRO SISTEMA YA DISPONIBLE

Las «BEE CARDS», módulos de memoria del tamaño de una tarjeta de crédito, tienen la capacidad de memoria de 256 K y ya están a la venta por parte de la compañía HUDSON SOFT, por £14.95.

Habrá tres tipos de BEE CARD. Cada una de ella contendrá un tipo de memoria distinto, según los programas.

El MASK ROM se usa para juegos de ordenador, publicidad electrónica y programas para promoción de ventas.

Las «CARDS» con EPROM se pueden usar para programas como archivos médicos, archivos de oficina y de clientes, donde la información puede escribirse en memoria.

Las «CARDS» EEPROM contienen memoria con posibilidad de ser reescrita y sirve básicamente para bases de datos.

En Japón, las «BEE CARDS» se están usando en conjunto con sistemas computerizados para comprar video juegos, compras por ordenador, reserva de hoteles y hasta para bancos caseros.

PROTECCION DE SOFTWARE

Como todos sabemos, cada día se desarrollan nuevos sistemas de protección de software para evitar el pirateo, y cada vez la gente intenta «destrozar» estos sistemas.

El último sistema «antipirateo» se llama LENSLOCK. Es un sistema desarrollado por la firma inglesa «ASAP Developments» y consiste en una rutina de código máquina que produce un código por pantalla, que debe ser tecleado antes de utilizar el programa. Además, junto con el software en cuestión, se suministra una lente con la cual, una vez que aparezca el código por pantalla la única manera de leer el mismo será mediante la lente.

Cada unidad de software vendrá acompañado por su respectiva lente, que sólo funcionará para ese juego, siendo imposible utilizarlo con otra lente, aunque sea de la misma compañía.

El primer juego que saldrá al mercado con este sistema de protección será una versión del programa ELITE, de la casa Firebird.

BUENAS NOTICIAS EN IMPRESORAS

RITEMAN ha sacado un nuevo modelo de impresora F+ de punto de matricial que ofrece impresión bidireccional de hasta 105 caracteres p.s., la opción de 96 caracteres itálicos ASCII y una cosa que pocas impresoras tienen: alimentación frontal de papel, lo que impide el típico lío entre papel y cables.

Asimismo, EPSON ha bajado el precio de sus impresoras RX100+ y FX80+ en £51 y £102 respectivamente.

VA DE JOYSTICK

SILICA SHOP, especialista en equipos de juegos de ordenador, acaba de importar algunos joystick americanos diseñados para hacer los juegos más fáciles de usar.

El Rapid Fire Control (control de disparo rápido), costará aproximadamente unas 2.000 ptas. y es un adaptador que se pone entre el puerto de joystick y el mismo joystick; con él, el jugador añade esta facilidad a sus juegos. Sólo puede usarse con aquel software que esté programado para permitir disparo rápido.

También existen cables alargados para joystick. Uno de ellos es de 12 pies y tiene forma de cable telefónico; su precio oscila alrededor de las 2.000 ptas., mientras que el de 6 pies costará aproximadamente unas 1.500 ptas.

Asimismo, los jugadores zurdos estarán encantados con el joystick South Paw y el adaptador de joystick para zurdos. Girando el mango del joystick 90°, el botón de disparo se sitúa en el lado derecho, para acomodarse a las necesidades de los zurdos.

Por último, el «Double Play» permite jugar con un joystick en un juego de dos jugadores y consiste en un cable en «y» que se enchufa, por un lado a un joystick y por otro lado a ambos puertos de joystick, lo que permite usar un solo joystick vía dos puertos.

INDEXADOS EN EL SPECTRUM

Todos sabemos que uno de los mayores inconvenientes con que nos encontramos a la hora de intentar hacer algo medianamente serio con nuestra maquinita, es que ésta no dispone de la posibilidad de trabajar con ficheros de datos y menos aún, la de acceder a un registro cualquiera de un bloque de éstos con un tiempo de repuesta mínimamente aceptable. Eso es lo que vamos a intentar a continuación, o mejor dicho, lo contrario, conseguir evitar eso..., en una palabra, vamos a hacerla mucho más operativa.

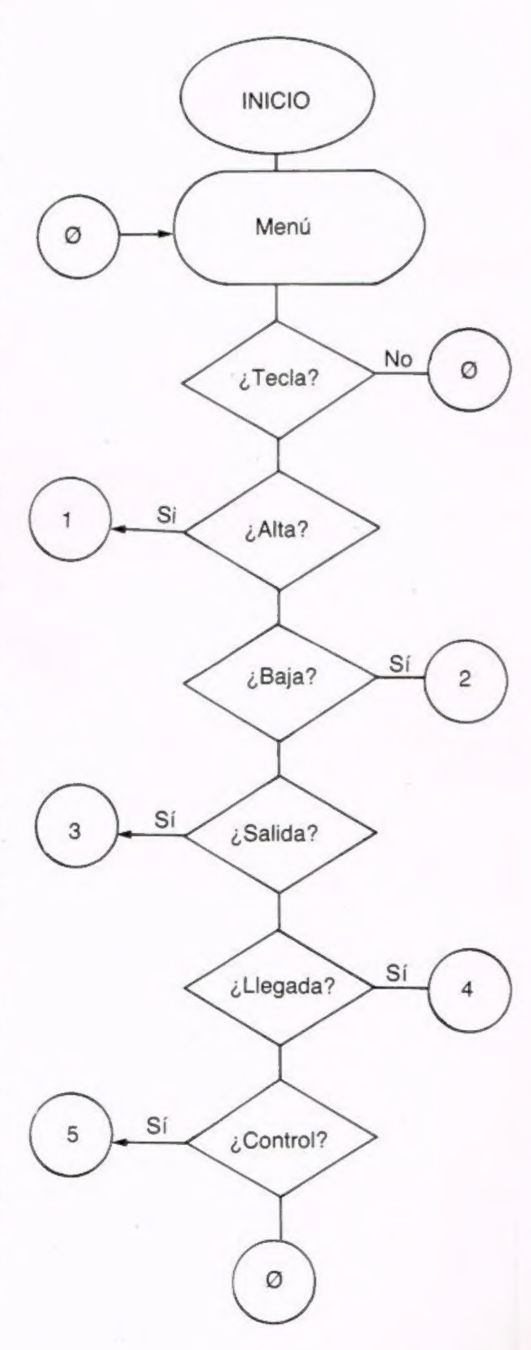
Un fichero indexado, tal y como se concibe en un ordenador profesional, nunca será posible, a menos que el número de datos a tratar fuera reducidísimo, pues el software que debiera de gestionar esto ocuparía la mayoría de la memoria de nuestro Spectrum. Pero no tenemos por qué desestimar nuestras posibilidades, ni mucho menos las de nuestro ordenador.

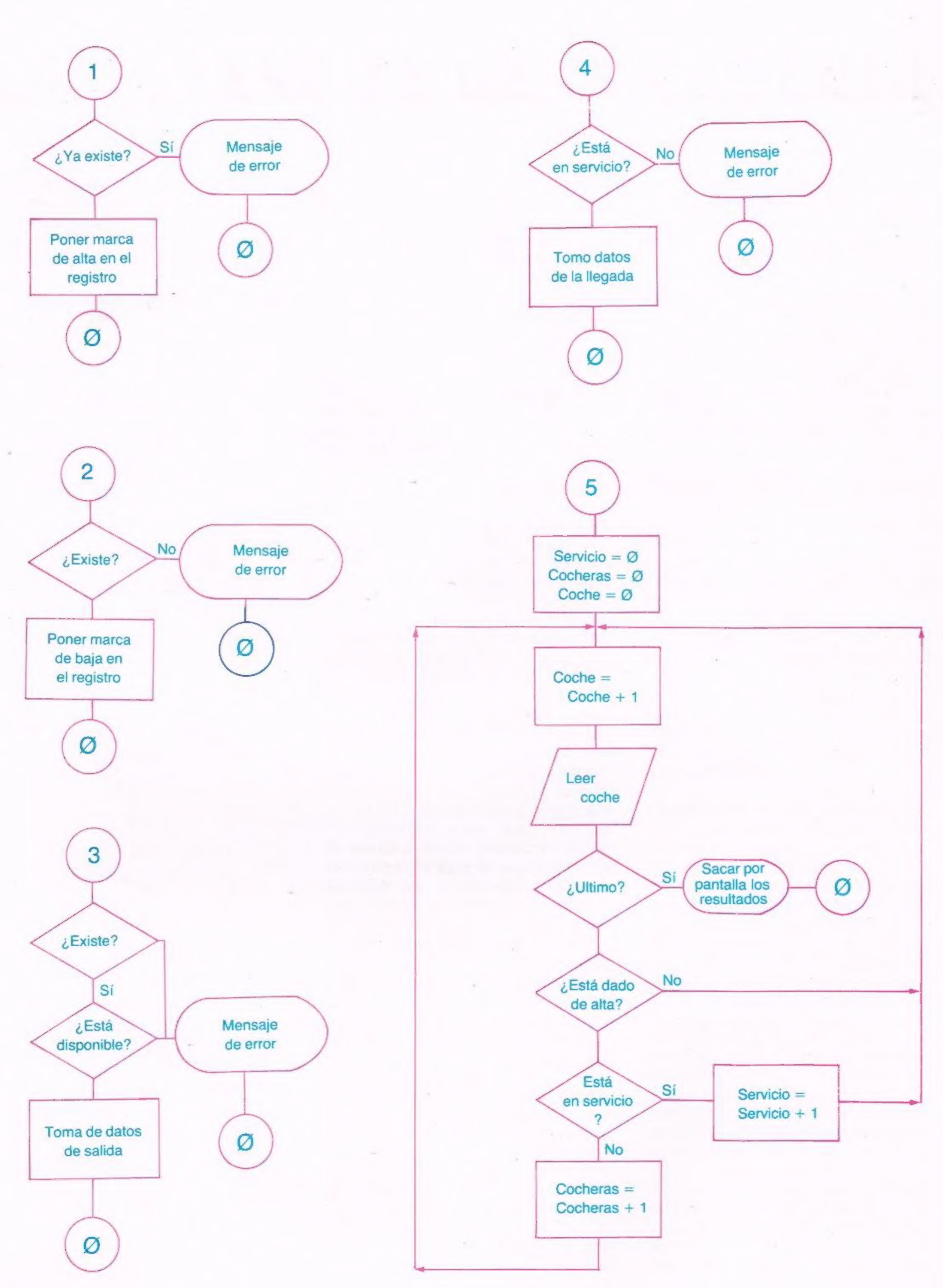
Estamos seguros de que para alguno de vosotros, la idea de un fichero indexado es totalmente nueva, por eso lo primero que vamos a hacer es tratar de explicarlo lo más fácilmente posible, para lo cual no consideramos necesario profundizar demasiado en el tema.

Imaginemos por un momento una estructura similar a la siguiente: escribiremos en un cuaderno todos los datos que quisiéramos tratar, por ejemplo los vehículos de una empresa de distribución de productos cualquiera, que lógicamente al llevar el número escrito sobre la carrocería no es fácil cambiarlo, por lo tanto a medida que la empresa vendió algunos vehículos, se iban quedando números desasignados. Una vez escritos todos los números y demás datos de cada vehículo, haremos una lista con todos los números de página y número mayor de vehículo escrito en esa página, lógicamente correspondidos; después haremos otras divisiones más pequeñas, por ejemplo a media página, con las que prepararemos otra lista con la página, media página y número mayor de coche escrito en ella, lógicamente correspondidos. Es de imaginar lo fácil que sería encontrar uno cualquiera de los vehículos referenciados, buscando en la primera lista el número inmediatamente superior y tomando el número de página, con el que buscaremos en la segunda lista que nos indicaría inmediatamente si se encuentra en el mitad superior o inferior de la misma.

Todo lo dicho anteriormente suena tan bien y tan fácil que en este momento seguro que te estarás preguntando por qué todos los ficheros no están organizados de esta manera, en respuesta a lo cual se nos presentan unos cuantos «contras», como, por ejemplo, ¿qué pasa en el caso de que compremos un vehículo nuevo y queramos aprovechar un número de los que ya vendimos...?, pues sencillísimo, se nos va al cuerno toda nuestra maravillosa organización. En los MainFrames esto se soluciona pudiendo insertar en cualquier momento cualquier dato o nuevo registro, y si por una entrada masiva de nuevos registros el área de índices se nos desborda, (esta área de índices, equivale a nuestras listas) existe un área de «overflow» o de desbordamiento, que..., pero creo que este es el momento de que paremos y nos dejemos de complicaciones, pues precisamente en este punto es donde comienzan las limitaciones de nuestro sistema. Como el tema de las listas consideramos que sería complicar las cosas demasiado, os presentamos un programa que localiza el registro en cuestión, partiendo de una clave de acceso, aplicándole una fórmula posteriormente que nos colocará en la variable índice, la dirección exacta donde empieza el registro que queríamos tratar; si a la dirección base obtenida, le aplicamos unos desplazamientos, nos podremos posicionar en cualquier campo dentro de nuestro registro. La única y mayor pega que tiene nuestro sistema, es que el fichero a crear debe de ser de longitud fija, el registro también y además cada registro ha de tener una clave fija, numérica y secuencial, es decir, partiendo del número uno hasta X, siendo X un número variable que dependerá de la longitud del registro que vayamos a crear y del espacio libre que nos deje el programa hecho por nosotros que gestionará esos datos.

A continuación os damos un programa, ...ese de que hablábamos antes, de control de vehículos, pero sólo para que os sirva de ejemplo de cómo se gestiona un programa de este tipo, también os ofrecemos unos diagramas de flujo para que os sea más fácil el seguimiento del mismo, no con demasidado detalle pero si con el suficiente como para que podáis entender la filosofía con que se construyó.




1
¿Ya existe?
Sí
Mensaje de error
Poner marca de alta en el registro
Ø
Ø
4
¿Está en servicio?
No
Mensaje de error
Tomo datos de la llegada
Ø
Ø
2
¿Existe?
No
Mensaje de error
Poner marca de baja en el registro
Ø
Ø
5
Servicio = Ø
Cocheras = Ø
Coche = Ø
Coche = Coche + 1
Leer coche
¿Ultimo?
Sí
Sacar por pantalla los resultados
Ø
¿Está dado de alta?
No
Está en servicio ?
Sí
Servicio = Servicio + 1
No
Cocheras = Cocheras + 1
3
¿Existe?
Sí
¿Está disponible?
Mensaje de error
Toma de datos de salida
Ø
Ø



MONSER, S.A.
Calle Argos, 9
28037 MADRID
Telf.: 742 72 12/96

ORDENA TU ORDENADOR



Ahora Vd. puede tener todo su equipo de ordenador en un gabinete de estilo con tres elegantes niveles. No más desórdenes de cables ni de periféricos. Además su equipo estará más protegido.

NO PIERDA ESTA OPORTUNIDAD UNICA

Tendrá espacio a su alcance para hardware y software.

Dispondrá de una unidad de puente de 56,5 cm ancho, 17 cm de alto y 30,5 cm de fondo para su televisor o monitor.

Debajo de esta unidad hay espacio suficiente para guardar su ordenador, aparato de cassette o microdrive.

En una tercera unidad tiene amplio espacio para guardar cintas, diskettes, joysticks, revistas, libros, etc.

Se vende desarmado en una caja plana, es muy fácil de armar, utilizando solamente una llave ALLEN.

El gabinete se presenta en dos colores, NOGAL y ROBLE y tiene dimensiones que se ajustan a las necesidades de espacio y altura que Vd. requiere.

ANCHO 85,5 cm. • ALTO 79,5 cm. • FONDO 60 cm.

Y ADEMAS LOS INTERFACES PARA SU JOYSTICK, IMPRESORA O MICRODRIVE

Interface DK'Tronics
Doble salida en la parte superior. La primera para joystick tipo Kempston y la segunda para software con teclas 6, 7, 8, 9 y 0 o redefinición de teclas.
Ref. 30001. P.V.P. 3.760 ptas.

Interface Centronic.
Para impresora y microdrive en paralelo. Ref. 30010.
P.V.P. 11.358 ptas.

Para pedidos simplemente rellene el cupón.

LOS JOYSTICKS DE GRAN RESPUESTA Y DURABILIDAD









Por favor, envíenme:(marco con una X):

	REF.	COLOR	CANTIDAD	PRECIO	SUBTOTAL
GABINETE	40005	Nogal		**8.975**	
	40005	Roble		**8.975**	
JOYSTICK	30007			**1.900**	
	30003			**2.598**	
	30008			**3.724**	
	30005			**2.450**	
INTERFACE	30001 30010			**3.760** **11.358**	

Gastos de envío gabinete 800 ptas., joysticks 200 ptas. ....

TOTAL ....

Forma de pago:

☐ Talón bancario a nombre de MONSER, S.A.
☐ Giro postal núm. ....
☐ Contra reembolso

Nombre ....

Domicilio ....

Ciudad.. Provincia ....

C.P. .... Telf.: ....

FIRMA

```
   5 POKE 23658,8
  10 POKE 23609,10: BORDER 0
  11 LET indice=35000
  13 INPUT "deme la fecha de hoy
          en formato DDMMAA :"
;d$
  15 LET i$=d$: GO SUB 500
  17 LET estado=1: LET cliente=2
: LET grupo=32: LET itinerario=5
2: LET lugarsal=112: LET horasal
=162: LET minusal=163: LET kilom
sal=164: LET kilomlleg=167: LET
kilomacum=170
  19 REM *** MENU ******
  20 CLS : PRINT "          OPCIO
NES MENU ",,," 1.- Salida coche
  "," 2.- Llegada coche
  "," 3.- Consulta coche        ","
4.- Estado general del parque  "
;" 5.- Cargar un estado  "," 6.-
 Guardar un estado "," 7.- Alta
vehiculo nuevo"," 8.- Baja vehic
ulo "
  30 INPUT FLASH 1;"Introduzca s
u opcion: ";opc: GO TO (3000+(op
c*300))
 500 REM ***control de fecha log
ica ****************************
*
 599 RETURN
1000 CLS : FOR n=1 TO 5: PRINT T
AB 2;"ESTE COCHE ESTA EN SERVICI
O": BEEP .5,25: NEXT n: PRINT :
PRINT : PRINT FLASH 1;"Pulse una
 tecla cualquiera": PAUSE 0: GO
TO 3300
1100 CLS : FOR n=1 TO 5: PRINT T
AB 2;"ESTE COCHE NO ESTA EN SERV
ICIO": BEEP .5,25: NEXT n: PRINT
 : PRINT : PRINT FLASH 1;"  Puls
e una tecla cualquiera": PAUSE 0
: GO TO 3600
1200 CLS : FOR n=1 TO 5: PRINT F
LASH 1;TAB 2;"ESTE COCHE YA EXIS
TE": BEEP .5,25: NEXT n: PRINT :
 PRINT : PRINT FLASH 1;"  Pulse
una tecla cualquiera": PAUSE 0:
GO TO 20
3300 REM **** Salida coche *****
3305 CLS : PRINT FLASH 1;"******
** SALIDA COCHE **********"
3310 INPUT FLASH 1;" Numero del
coche?? ";coche
3314 IF coche=.5 THEN GO TO 20
3315 IF coche<1 OR coche>70 THEN
 GO TO 3310
3320 LET posic=indice+(200*(coch
e-1))
3322 IF PEEK (posic+estado)<>2 T
HEN GO TO 1000
3325 POKE posic,coche: PRINT "Co
che: ";coche;AT 1,15;"■";
3330 POKE (posic+estado),1
3332 INPUT "fecha del servicio?
";f$: LET d$=f$: GO SUB 500
3335 PRINT "fecha: ";f$( TO 2);"
/";f$(3 TO 4);"/";f$(5 TO 6)
3340 POKE (posic+173),VAL f$( TO
 2): POKE (posic+174),VAL f$(3 T
O 4): POKE (posic+175),VAL f$(5
TO 6)
3345 LET cli=posic+cliente: INPU
T "Cliente? ",,m$: LET c$=m$: IF
 LEN m$>29 THEN LET c$=m$(1 TO 2
9)
3346 LET lon=LEN c$: POKE cli,lo
n: FOR n=1 TO lon: POKE (cli+n),
CODE c$(n TO n): NEXT n
3347 PRINT "CLIENTE
     ": PRINT c$
3350 LET gru=posic+grupo: INPUT
"Grupo? ",,g$: LET c$=g$: IF LEN
 g$>19 THEN LET c$=g$(1 TO 19)
3353 LET lon=LEN c$: POKE gru,lo
n: FOR n=1 TO lon: POKE (gru+n),
CODE c$(n TO n): NEXT n
3355 PRINT "GRUPO
     ": PRINT c$
3360 LET iti=posic+itinerario: I
NPUT "Itinerario ? ",,y$: LET c$
=y$: IF LEN Y$>59 THEN LET c$=y$
(1 TO 59)
3363 LET lon=LEN c$: POKE iti,lo
n: FOR n=1 TO lon: POKE (iti+n),
CODE c$(n TO n): NEXT n
3365 PRINT TAB 0;"ITINERA
RIO     ";c$
3370 LET lus=posic+lugarsal: INP
UT "Lugar de salida ? ",y$: LET
c$=y$: IF LEN Y$>49 THEN LET c$=
y$(1 TO 49)
3373 LET lon=LEN c$: POKE lus,lo
n: FOR n=1 TO lon: POKE (lus+n),
CODE c$(n TO n): NEXT n
3375 PRINT TAB 0;"LUGAR DE SA
LIDA     ";c$
3380 LET hsa=posic+horasal: INPU
T "Hora de salida ?";hs: IF hs>2
4 THEN GO TO 3380
3382 POKE hsa,hs: PRINT "Hora "
;hs;",";
3385 LET msa=posic+minusal: INPU
T "Minuto de salida ?";ms: IF ms
>59 OR ms<0 THEN GO TO 3385
3387 POKE msa,ms: PRINT ms
3390 LET ksa=posic+kilomsal: INP
UT "Kms. salida? ";ks
3392 PRINT "Kms salida ";ks
3394 LET hi=INT (ks/65536): POKE
 ksa,hi: LET ks=ks-(hi*65536)
3396 LET hi=INT (ks/256): POKE (
ksa+1),hi: LET ks=ks-(hi*256)
3398 POKE (ksa+2),ks
3499 PRINT AT 21,0; FLASH 1;"
     PULSE UNA TECLA          ":
PAUSE 0: GO TO 20
3600 REM **** llegada coche ****
*
3605 CLS : PRINT FLASH 1;"******
** LLEGADA COCHE *********"
3610 INPUT FLASH 1;" Numero del
coche?? ";coche
3614 IF coche=.5 THEN GO TO 20
3615 IF coche<1 OR coche>70 THEN
 GO TO 3610
3620 LET posic=indice+(200*(coch
e-1))
3622 IF PEEK (posic+estado)<>1 T
HEN GO TO 1100
3625 BEEP .01,50: PRINT "Coche:
";coche;AT 1,15;"■";
3630 POKE (posic+estado),2
3632 BEEP .01,50: PRINT "F.Serv.
 ";PEEK (posic+173);"/";PEEK (po
sic+174);"/";PEEK (posic+175)
3636 PRINT "CLIENTE
     "
3640 LET cli=posic+cliente: FOR
n=1 TO PEEK cli: LET carac=PEEK
(cli+n): BEEP .001,50: PRINT CHR
$ carac;: NEXT n
3645 PRINT TAB 0;"GRUPO
     "
3650 LET gru=posic+grupo: FOR n=
1 TO PEEK gru: LET carac=PEEK (g
ru+n): BEEP .001,20: PRINT CHR$
carac;: NEXT n
3655 PRINT TAB 0;"ITINERA
RIO     "
3660 LET iti=posic+itinerario: F
OR n=1 TO PEEK iti: LET carac=PE
EK (iti+n): BEEP .001,50: PRINT
CHR$ carac;: NEXT n
3665 PRINT TAB 0;"LUGAR DE SA
LIDA     "
3670 LET LUS=posic+lugarsal: FOR
 n=1 TO PEEK lus: LET carac=PEEK
 (lus+n): BEEP .001,20: PRINT CH
R$ carac;: NEXT n
3675 PRINT TAB 0;"______________
___________"
3680 LET hsa=posic+horasal: PRIN
T TAB 0;"Hora de salida ";PEEK
 hsa;",";PEEK (hsa+1): BEEP .01,
50
3685 PRINT "                _____
_______________"
3690 LET ksa=posic+kilomsal: LET
 a=PEEK ksa: LET b=PEEK (ksa+1):
 LET c=PEEK (ksa+2)
3692 PRINT "  Kms salida ";(
65536*a)+(256*b)+c
3695 INPUT "Kms. de llegada del
servicio ? ";kll: PRINT "  Kms
 llegada ";kll
3697 LET klle=posic+kilomlleg
3699 LET hi=INT (kll/65536): POK
E klle,hi: LET kll=kll-(hi*65536
)
3700 LET hi=INT (kll/256): POKE
(klle+1),hi: LET kll=kll-(hi*256
)
3705 POKE (klle+2),kll
3710 LET kkmsal=(PEEK ksa*65536)
+(PEEK (ksa+1)*256)+PEEK (ksa+2)
3715 LET kkmlle=(PEEK klle*65536
```

```
)+(PEEK (klle+1)*256)+PEEK (klle
+2)
3720 LET kkmrec=kkmlle-kkmsal: P
RINT "  kms. hechos  ";kkmrec
3725 LET pkmac=posic+kilomacum:
LET kmac=(PEEK pkmac*65536)+(PEE
K (pkmac+1)*256)+PEEK (pkmac+2):
 LET kmac=kmac+kkmrec
3730 PRINT "Este coche lleva acu
mulados con la empresa, kms. ";k
mac: PRINT "________________________________
"
3735 LET hi=INT (kmac/65536): PO
KE pkmac,hi: LET kmac=kmac-(hi*6
5536)
3737 LET hi=INT (kmac/256): POKE
 (pkmac+1),hi: LET kmac=kmac-(hi
*256)
3739 POKE (pkmac+2),kmac
3750 PRINT AT 21,0; FLASH 1;"
     PULSE UNA TECLA          ":
PAUSE 0: GO TO 20
3900 REM ****consulta coche ***
**
3905 CLS : PRINT FLASH 1;"******
** CONSULTA COCHE ********"
3910 INPUT FLASH 1;" Numero del
coche?? ";coche
3914 IF coche=.5 THEN GO TO 20
3915 IF coche<1 OR coche>70 THEN
 GO TO 3910
3920 LET posic=indice+(200*(coch
e-1))
3925 BEEP .01,50: PRINT "Coche:
";coche;AT 1,15;" ";
3932 BEEP .01,50: PRINT "F.Serv.
 ";PEEK (posic+173);"/";PEEK (po
sic+174);"/";PEEK (posic+175)
3938 PRINT "C L I E N T E
          "
3940 LET cli=posic+cliente: FOR
n=1 TO PEEK cli: LET carac=PEEK
(cli+n): BEEP .001,50: PRINT CHR
$ carac;: NEXT n
3945 PRINT TAB 0;"G R U P O
             "
3950 LET gru=posic+grupo: FOR n=
1 TO PEEK gru: LET carac=PEEK (g
ru+n): BEEP .001,20: PRINT CHR$
carac;: NEXT n
3955 PRINT TAB 0;"I T I N E R A
R I O            "
3960 LET iti=posic+itinerario: F
OR n=1 TO PEEK iti: LET carac=PE
EK (iti+n): BEEP .001,50: PRINT
CHR$ carac;: NEXT n
3965 PRINT TAB 0;"LUGAR  DE   S A
L I D A          "
3970 LET LUS=posic+lugarsal: FOR
 n=1 TO PEEK lus: LET carac=PEEK
 (lus+n): BEEP .001,20: PRINT CH
R$ carac;: NEXT n
3975 PRINT TAB 0;"________________
________________"
3980 LET hsa=posic+horasal: PRIN
T TAB 0;"Hora de salida  ";PEEK
 hsa;",";PEEK (hsa+1): BEEP .01,
50
3985 PRINT "                 ____
________________"
3986 IF PEEK (posic+estado)=1 TH
EN PRINT FLASH 1;"  ESTE COCHE
ESTA DE SERVICIO  ": GO TO 4000
3987 IF PEEK (posic+estado)=2 TH
EN PRINT FLASH 1;"  ESTOS DATOS
CORRESPONDEN AL --ULTIMO SERVICI
O REALIZADO POR --ESTE COCHE.
               "
3990 LET ksa=posic+kilomsal: LET
 a=PEEK ksa: LET b=PEEK (ksa+1):
 LET c=PEEK (ksa+2)
3992 PRINT "  Kms. salida  ";(
65536*a)+(256*b)+c
3995 LET kll=(PEEK (posic+kiloml
leg)*65536)+(PEEK (posic+kilomll
eg+1)*256)+PEEK (posic+kilomlleg
+2): PRINT "  Kms. llegada  ";k
ll
4000 LET pkmac=posic+kilomacum:
LET kmac=(PEEK pkmac*65536)+(PEE
K (pkmac+1)*256)+PEEK (pkmac+2)
4005 PRINT "Este coche lleva acu
mulados con la empresa, kms. ";k
mac: PRINT "________________________________
"
4050 PRINT AT 21,0; FLASH 1;"
     PULSE UNA TECLA          ":
PAUSE 0: GO TO 20
```

```
4200 REM ****estado general del
parque ***************************
4205 CLS : PRINT AT 7,5; FLASH 1
;"Submenu Estado": PRINT : LET t
=3: PRINT TAB t;"a.- Coches en s
ervicio";TAB t;"b.- Coches en co
chera";TAB t;"c.- Totales numeri
cos"
4210 INPUT "introduzca su opcion
 : ";o$: GO TO (4200+CODE o$)
4265 GO TO 4280
4266 GO TO 4300
4267 GO TO 4320
4280 CLS : FOR n=1 TO 70: LET po
sic=indice+(200*(n-1)): IF PEEK
(posic+estado)=1 THEN PRINT "Coc
he numero ";PEEK posic;TAB 17;"
en servicio"
4281 NEXT n
4282 PRINT FLASH 1;"   PULSE CUA
LQUIER TECLA          ": PAUSE 0:
GO TO 20
4300 CLS : FOR n=1 TO 70: LET po
sic=indice+(200*(n-1)): IF PEEK
(posic+estado)=2 THEN PRINT "Coc
he numero ";PEEK posic;TAB 17;"
en cocheras"
4301 NEXT n
4302 PRINT FLASH 1;"   PULSE CUA
LQUIER TECLA          ": PAUSE 0:
GO TO 20
4320 CLS : LET par=0: LET coc=0:
 LET ser=0: FOR n=1 TO 70: LET p
osic=indice+(200*(n-1)): IF PEEK
 (posic+estado)=1 THEN LET ser=s
er+1: LET par=par+1: GO TO 4330
4322 IF PEEK (posic+estado)=2 TH
EN LET coc=coc+1: LET par=par+1:
 GO TO 4330
4330 NEXT n
4335 CLS : PLOT 60,50: DRAW 0,80
: DRAW 135,0: DRAW 0,-80: DRAW -
135,0
4340 PRINT AT 7,8;"en cocheras :
 ";coc;AT 9,8;"en servicio : ";s
er;AT 12,8;"TOTAL parque: ";par
4350 PRINT AT 21,0; FLASH 1;"
     PULSE UNA TECLA          ":
PAUSE 0: GO TO 20
4500 REM ****cargar un estado***
4510 CLS : PRINT AT 5,0; FLASH 1
;"  PONGA EN MARCHA EL CASSETTE
  ": LOAD ""CODE indice,14000
4520 GO TO 20
4800 REM *** SALVAR un estado***
4810 CLS : INPUT "DAME LA FECHA
DE HOY EN DD/MM/AA";F$: PRINT AT
 5,0; FLASH 1;" PONGA EN MARCHA
EL CASSETTE EN MODO GRABACION Y
PULSE UNA TECLA": SAVE F$CODE in
dice,14000
4820 GO TO 20
5100 REM **** alta coche *****
5110 CLS : PRINT AT 5,0; FLASH 1
;"ESTAS DANDO DE ALTA UN VEHICUL
O ": INPUT "numero del coche? ";
coche: IF COCHE=.5 THEN GO TO 20
5111 LET posic=indice+(200*(coch
e-1)): CLS
5112 IF PEEK (posic+1)<>0 THEN G
O TO 1200
5115 POKE (posic+1),2: POKE posi
c,coche
5120 GO TO 20
5400 REM ****BAJA coche *****
5410 CLS : PRINT AT 5,0; FLASH 1
;"ESTAS DANDO DE BAJA UN VEHICUL
O": INPUT "numero del coche? ";
coche: IF COCHE=.5 THEN GO TO 20
5411 LET posic=indice+(200*(coch
e-1)): CLS
5412 IF PEEK (posic+1)=0 THEN BE
EP 3,50: PRINT FLASH 1;"ESTE COC
HE NO SE PUEDE DAR BAJA ESTE COC
HE NO SE PUEDE DAR BAJA ESTE COC
HE NO SE PUEDE DAR BAJA
                              PUL
SA CUALQUIER TECLA         : PAUSE
 0: GO TO 5410
5415 POKE (posic+1),0
5420 GO TO 20
9999 PRINT PEEK 23653+256*PEEK 2
3654-24000
```

JUEGO DEL MES

Frank N. Stein

Corría el año 1884 cuando el profesor FRANK N. STEIN había organizado su laboratorio en lo más recóndito del Bosque Negro con la extraña intención de convertirse en algo así como un dios, haciendo algo que sólo quien ostenta esa calidad, puede hacerlo..., crear vida, dotar de vida, de espíritu, a un ser creado por sus propias manos uniendo trocitos recogidos de uno y otro sitio. Pero..., el Doctor Stein sufre de una terrible gripe que le ha paralizado los miembros inferiores y necesita urgentemente de un ayudante lo bastante atrevido como para que le guíe en su terrible misión de reconstruir su monstruo.

Pieza a pieza, el Dr. Stein tendrá que formar el esqueleto de su mítica figura, para después, con la energía eléctrica transformada, dar vida al fantástico ser, que se volverá contra él en la siguiente pantalla, donde lógicamente tendremos que guiar al Dr. Stein para que destruya su obra para volver a intentar crearla, esta vez con mucha más dificultad, y así hasta cincuenta pantallas diferentes, cada vez mucho más difíciles; pero vayamos por partes.

Como suponemos que habrás aceptado el reto, queremos informarte de algunas cosas que seguramente necesitarás saber y de paso darte algunos consejos que te vendrán al pelo.

En primer lugar vamos a cargar el juego. Rebobina la cinta hasta su comienzo, teclea LOAD"" seguido de «ENTER» y a continuación pulsa PLAY en el cassette. El programa se compone de tres partes claramente diferenciadas entre sí, el cargador en Basic, la pantalla de presenta-



ción, que nos mostrará durante algunos minutos la figura del doctor, eso sí, bastante ampliada, y a continuación entrará el juego en sí, realizado íntegramente en código máquina.

Las teclas de movimiento son:

Z para la izquierda
X para la derecha
ESPACE PARA SALTAR.



Una pequeña sutileza que no estará de más el comentarla es que para saltar no basta con pulsar la tecla «space», sino que igualmente será necesario colocar al Dr. Stein sobre los muelles que a tal efecto se encuentran distribuidos por la pantalla.

Vamos a analizar las dos primeras pantallas que nos servirán de base para todas las demás.

En la primera pantalla encontramos repartidos por varios niveles los trocitos de esqueleto que tenemos que ir tomando, pero en el orden exacto, es decir, empezando por la cabeza (es decir, por la calavera), después el hombro derecho, luego el izquierdo, la cadera derecha, a continuación la izquierda, etc., así hasta que como decíamos anteriormente completemos el cuerpo de nuestro ser; en ese momento será necesario volver a subir al lado de nuestra obra y apretar el mecanismo que dotará de vida al monstruo, momento en que pasaremos a la siguiente pantalla. En ésta, nuestro «enemigo» nos asediará arrojándonos ruedas; del mismo modo habremos de eludir montones de peligros menores antes de llegar al piso superior, donde la misma electricidad que dio vida a nuestra criatura se la quitará, para pasar de nuevo a otra habitación laboratorio del castillo del Bosque Negro, a recomenzar la obra.

Hay en la habitación del laboratorio unos puntos importantes a tener en cuenta, son las derivaciones de energía eléctrica con las que podemos toparnos en unos topecitos metálicos repartidos por los pasillos. Otro detalle curioso es el de la barra que hay en el extremo derecho de la pantalla y que si pasamos sobre ella pulsando la tecla «Space», nos permitirá caer al piso inferior.

Por último sólo nos queda recomendaros que antes de comenzar a moveros en cualquier pantalla, la estudiéis detenidamente.

¡Buena suerte!

CODIGO MAQUINA

CAPITULO VIII
(Continuación)

4.—Bifurcaciones relativas

En el capítulo anterior veíamos cómo podíamos bifurcar a un lugar fijo de la memoria. Dicha dirección estaba representada por un número, o por un nombre simbólico. Si queríamos trasladar la rutina a otro lugar de la memoria debíamos cambiar el número y ensamblar de nuevo la rutina. Para permitir la transportabilidad de una rutina desde una posición de memoria a otra, sin necesidad de ensamblar de nuevo, el microprocesador tiene otras instrucciones de salto, o bifurcación, que además, como veremos tienen una característica común de ocupar menos espacio que sus gemelas.

5.—Salto incondicional relativo

El formato de esta instrucción en Assembler es:

JR d

en donde,

d.— es un valor comprendido entre − 128 y + 127 que representa el desplazamiento.

La traducción a Código Máquina de esta instrucción es:

18 xx

El funcionamiento de esta instrucción es sencillo. Se limita a sumar o restar el valor de la constante d al registro contador de programa PC una vez que éste ya ha sido incrementado durante la ejecución de esta instrucción, y por lo tanto el PC en ese momento señala a la siguiente. De esto se deduce que una instrucción JR Ø no hace absolutamente nada.

Si se utiliza un programa ensamblador, generalmente no es necesario tener en cuenta los desplazamientos, pues si en lugar de la constante d colocamos el nombre simbólico de la instrucción a la que vamos a bifurcar, el programa se encargará de hacer los cálculos. Para aquellos que no dispongan de esa utilísima herramienta a continuación le ofrecemos una tabla que permite el cálculo de los números negativos con su traducción a hexadecimal.

Junto a la tabla, y por si alguno prefiere tenerla en el ordenador, ofrecemos el programa, en Basic, con el que la hemos obtenido.

SEGUNDO SEMIBYTE

PRIMER SEMIBYTE

	0	1	2	3
8	−128	−127	−126	−125
9	−112	−111	−110	−109
A	−96	−95	−94	−93
B	−80	−79	−78	−77
C	−64	−63	−62	−61
D	−48	−47	−46	−45
E	−32	−31	−30	−29
F	−16	−15	−14	−13

	4	5	6	7
8	−124	−123	−122	−121
9	−108	−107	−106	−105
A	−92	−91	−90	−89
B	−76	−75	−74	−73
C	−60	−59	−58	−57
D	−44	−43	−42	−41
E	−28	−27	−26	−25
F	−12	−11	−10	−9

	8	9	A	B
8	−120	−119	−118	−117
9	−104	−103	−102	−101
A	−88	−87	−86	−85
B	−72	−71	−70	−69
C	−56	−55	−54	−53
D	−40	−39	−38	−37
E	−24	−23	−22	−21
F	−8	−7	−6	−5

	C	D	E	F
8	−116	−115	−114	−113
9	−100	−99	−98	−97
A	−84	−83	−82	−81
B	−68	−67	−66	−65
C	−52	−51	−50	−49
D	−36	−35	−34	−33
E	−20	−19	−18	−17
F	−4	−3	−2	−1

6.—Saltos condicionales relativos

También este conjunto de instrucciones permite realizar una ruptura de control cuando se cumpla una condición de las que queda constancia en el registro de flags (F).

El formato de estas instrucciones es similar al de las bifurcaciones absolutas condicionales. Es decir,

JR cc,d

en donde,

cc.— es el indicativo de la condición que debe haberse cumplido para que se produzca la ruptura de control y se efectúe la bifurcación.

d.— es un valor comprendido entre −128 y +127 que representa el desplazamiento o cantidad que hay que sumar al registro PC para obtener la dirección de la siguiente instrucción a ejecutarse.

Como en las bifurcaciones condicionales que vimos en el capítulo anterior, también en este caso se trata de un conjunto de instrucciones que vamos a incluir en una tabla conjuntamente con su traducción a Código Máquina.

Assembler	Código máquina
JR C, des	38 xx
JR NC, des	30 xx
JR Z, des	28 xx
JR NZ, des	20 xx

Como puede apreciarse, el número de instrucciones de bifurcaciones relativas condicionales es mucho menor que el de las bifurcaciones absolutas. No obstante, éstas son las utilizadas, como podéis apreciar en cualquier programa que veáis escrito en Assembler.

También observamos que todas ellas ocupan dos bytes de memoria en lugar de los tres que ocupan sus hermanas.

7.—Ejercicio

En esta ocasión, y dado que ya hemos realizado una muy larga y algo compleja rutina en ocasión del capítulo anterior, preferiría aprovechar la misma, para que haciendo unas pequeñas modificaciones podamos conseguir unas rutinas totalmente «reubicables», que en realidad quiere decir que las podamos poder donde queramos sin necesidad de ensamblarlas más que en el primer programa en que las utilicemos.

La primera modificación que se me ocurre es sustituir la instrucción número 60 por JR TEXT1, ya que la instrucción donde figura el nombre simbólico (la 30) se encuentra a una distancia inferior a −128 bytes, (9 bytes de las CALL y 2 ó 3 por la JR o JP).

La segunda modificación, y por el mismo motivo que el anterior, la línea 140 la sustituiría por:

JR SABAS

Por último, dentro de la rutina TECLA podemos sustituir las instrucciones 320 y 340 por:

```
320   JR   NZ,TECLN
340   JR   NC,TECLN
```

Con estos cambios, además de lo anecdótico de habernos ahorrado cuatro bytes, lo que conseguimos es que la rutina TECLA, una vez ensamblada podamos moverla a cualquier posición de memoria. Esto ya no es sólo una anécdota.

```
   1 REM    Tabla de numeros negativos          Por A. Barrios
  10 LET t$="0123456789ABCDEF"
  20 FOR z=0 TO 13 STEP 4
  30 FOR y=1 TO 4: PRINT AT 1,1+y*6;t$(y+z): NEXT y
  40 FOR x=8 TO 15: PRINT : PRINT TAB 1;t$(x+1);: FOR y=0 TO 3
  50 LET a=16*x+y+z-256
  60 PRINT TAB 2+(y+1)*6-LEN STR$ a;a;" ";
  70 NEXT y: PRINT : NEXT x
  80 PLOT 0,175  DRAW 255,0: DRAW 0,-175: DRAW -255,0: DRAW 0,175
  90 PLOT 0,155: DRAW 255,0
 100 PLOT 20,0. DRAW 0,175
 110 COPY : PAUSE 0
 120 CLS : NEXT z
```

MONSER, S.A.

Con motivo del reciente acuerdo, para la comercialización en nuestro país del TECLADO PROFESIONAL para SPECTRUM, te ofrecemos, en exclusiva, esta SUPER OFERTA sin igual hasta la fecha.

6.500 pts.



CARACTERISTICAS DEL TECLADO

- Construido con material ABS de alta densidad.
- Panel numérico separado.
- Conexiones accesibles por la parte trasera.
- Espacio interior para colocar numerosos periféricos (Microdrive, Transformador de corriente, etc.).
- Tecla "DELETE" de utilización directa.
- Teclas con funciones impresas de fábrica.

Super oferta:

Junto a la suscripción a cualquiera de nuestras revistas y adquiriendo el teclado profesional o Joystick e Interface, te ofrecemos estos increíbles precios.

Haz cuentas y verás cómo realmente merece la pena.

Conviértete y convierte a tu Spectrum en un profesional.

6.500 pts.



INTERFACE

- Con doble salida.
- La primera salida simula las teclas 6, 7, 8, 9 y 0.
- La segunda salida, funciona con juegos compatibles con Kempston.



JOYSTICK

- Mango con diseño anatómico.
- Posibilidades de disparo automático.
- Dos teclas de disparo.
- Ventosas super-adherentes.



+ dk'tronics®



Software Magazine.
Cada número consta de dos cassettes con programas estrella, más la revista de pasatiempos JAKEKA. Aparición mensual.

11.525 ptas.



48 K.
La mejor revista de Micro Informática con cassette, al mejor precio. Aparición mensual, coleccionable.

8.375 ptas.



Ordenador Educativo.
La primera revista educativa para Spectrum. Contiene un cassette con tres programas educativos, con los que sus hijos se divertirán aprendiendo.

9.455 ptas.

MONSER, S.A. C / Argos, 9. 28037 MADRID.

DESEO QUE ME ENVIEN:
Marco con una (X) en el casillero correspondiente

☐ Teclado profesional y suscripción a *S. Magazine*
☐ Joystick e Interface y suscripción a *S. Magazine* .... 11.525 ptas.

☐ Teclado profesional y suscripción a *O. Educativo*
☐ Joystick e Interface y suscripción a *O. Educativo* .... 9.455 ptas.

☐ Teclado profesional y suscripción a *48 K*
☐ Joystick e Interface y suscripción a *48 K* .... 8.375 ptas.

☐ Teclado profesional .... 6.500 ptas.

☐ Joystick e Interface .... 6.500 ptas.

La suscripción será de 12 números a partir del N.º ....

NOMBRE........ EDAD ....
APELLIDOS ....
DOMICILIO ....
CIUDAD........ PROVINCIA ....
C. POSTAL........ TELF. ....

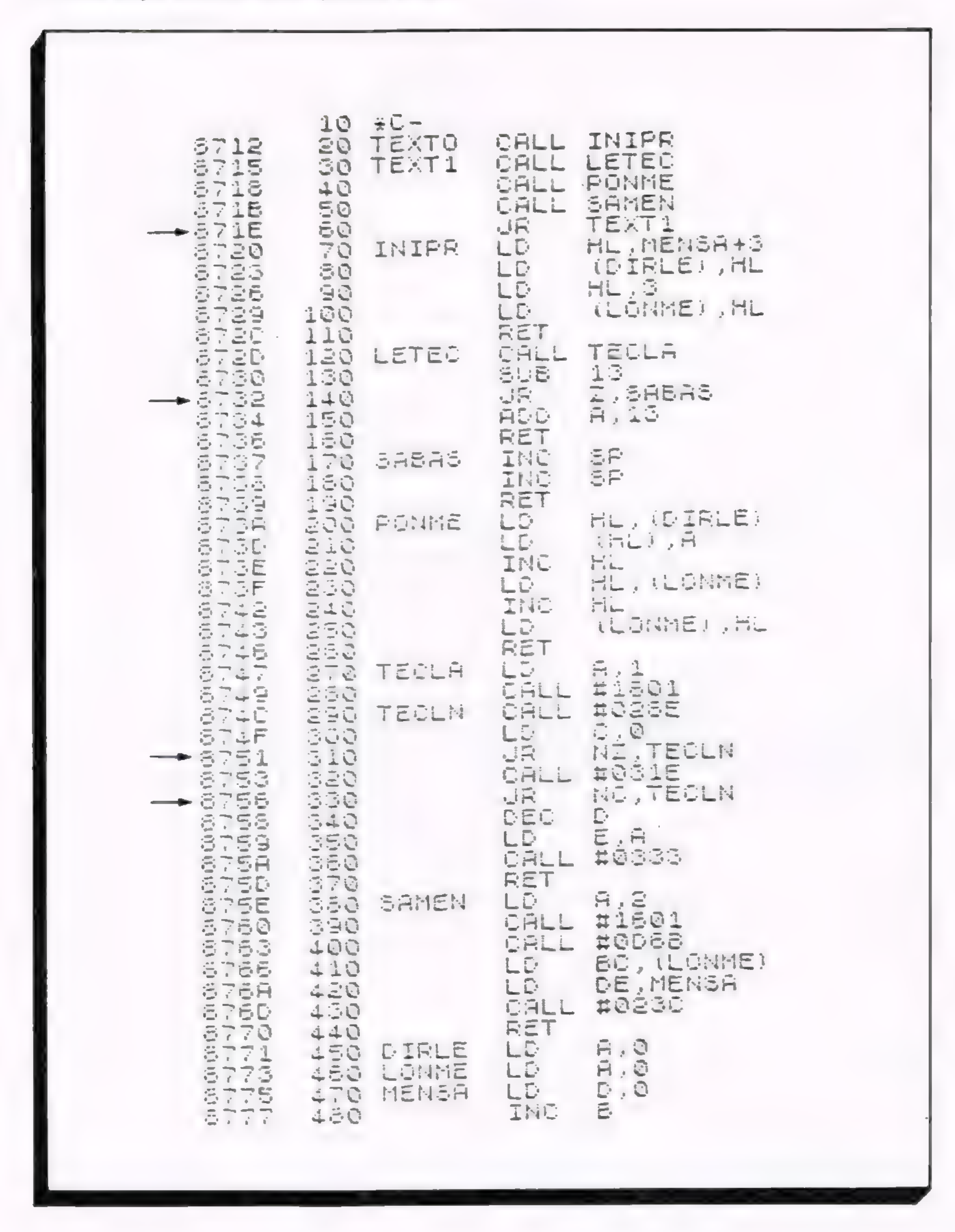
Marco con una (X) en el casillero correspondiente la forma de pago que más me conviene

☐ Talon bancario adjunto a nombre de MONSER, S.A.
☐ Giro Postal N.º ....
☐ Contra reembolso del primer numero

Total ....

FIRMA

La rutina ya modificada quedará así:



```
            *C-
8712  TEXTO   CALL  INIPR
8715  TEXT1   CALL  LETEC
8718          CALL  PONME
871B          CALL  SAMEN
871E          JR    TEXT1
8720  INIPR   LD    HL,MENSA+3
8723          LD    (DIRLE),HL
8726          LD    HL,3
8729          LD    (LONME),HL
872C          RET
872D  LETEC   CALL  TECLA
8730          SUB   13
8732          JR    Z,SABAS
8734          ADD   A,13
8736          RET
8737  SABAS   INC   SP
8738          INC   SP
8739          RET
873A  PONME   LD    HL,(DIRLE)
873D          LD    (HL),A
873E          INC   HL
873F          LD    HL,(LONME)
8742          INC   HL
8743          LD    (LONME),HL
8746          RET
8747  TECLA   LD    A,1
8749          CALL  #1601
874C  TECLN   CALL  #028E
874F          LD    C,0
8751          JR    NZ,TECLN
8753          CALL  #031E
8756          JR    NC,TECLN
8758          DEC   D
8759          LD    E,A
875A          CALL  #0333
875D          RET
875E  SAMEN   LD    A,2
8760          CALL  #1601
8763          CALL  #0D6B
8766          LD    BC,(LONME)
876A          LD    DE,MENSA
876D          CALL  #203C
8770          RET
8771  DIRLE   LD    A,0
8773  LONME   LD    A,0
8775  MENSA   LD    D,0
8777          INC   B
```

CAPITULO IX

1.—INTRODUCCION

En este pequeño capítulo vamos a ver unas pseudoinstrucciones, que son muy utilizadas en Assembler, pero que no dicen nada al microordenador, y también, relacionado con esto, veremos la instrucción que no hace nada.

2.—Directivos de ensamblador

Se denominan así algunas falsas instrucciones, que se intercalan entre las verdaderas, pero que no se traducirán en órdenes al microprocesador, sino que realmente son órdenes al propio programa traductor.

Todos los ensambladores tienen un juego más o menos completo de directivos y por ello es común contemplarlos en aquellos programas que aparecen en revistas y libros. Aunque este conjunto de instrucciones es muy propio de cada traductor vamos a tratar de contemplar los más generales, ya que los específicos del traductor vienen explicados en las propias instrucciones del programa.

2.1. Directivo ORG

El formato de este directivo es:

ORG expresión.

Esta instrucción al Ensamblador le indica que, a partir del momento en que se encuentra dicha orden, el resultado de la traducción (el programa en código máquina) lo almacene en la dirección indicada en «expresión».

El valor, o formato, que puede tener «expresión» depende fundamentalmente de la potencia del traductor. Los hay que únicamente admiten un número, ya sea en decimal o hexadecimal, y también los hay que admiten una expresión matemática, una fórmula, con sus relaciones aritméticas de sumar, restar, multiplicar y dividir.

Por ejemplo, si intercalamos la instrucción:

ORG 30000

le indicaremos al Ensamblador que, desde este momento, todas las instrucciones que vienen a continuación las traduzca y su resultado lo sitúe a partir de la posición 30000 de memoria.

Como ejercicio, podríamos intercalar un directivo de este tipo entre dos instrucciones de alguna rutina que ya habíamos visto.

Como observación importante hemos de señalar que no debemos olvidar en qué posición están nuestras otras rutinas, o el propio programa ensamblador, pues puede ocurrir que si damos una dirección no apropiada destruyamos una zona de memoria que pudiera tener interés para nosotros.

No obstante, esta instrucción suele ponerse al comienzo del programa, y no hacerse modificaciones posteriores. De este modo se evitan

errores de manipulación de direcciones.
Esta instrucción no lleva nombre simbólico.

2.2. Directivo EQU

Este directivo tiene el formato

nombre EQU expresión

La función del mismo es igualar un nombre a un valor. Por ejemplo, si decimos:

PP EQU 25

indicaremos al programa traductor, ensamblador, que cada vez que vea la expresión PP en algún operando la traduzca por el valor 25.
En este directivo, como es fácil suponer, es siempre obligatorio dar un «nombre» a la instrucción. Es función de la potencia del traductor considerar la complejidad que puede expresarse dentro de la palabra «expresión».

2.3. Directivo DEFB

El formato de esta intrucción es:

DEFB expresión

En general, este directivo sirve para reservar una zona de memoria de *un solo byte,* en el que se almacena el valor binario de «expresión».

Generalmente suelen llevar un nombre simbólico por el cual se va a conocer dicho campo durante todo el programa.

2.4. Directivo DEFW

El formato de esta instrucción es:

DEFW expresión

En general, este directivo sirve para reservar una toma de memoria, de *dos bytes* de longitud, y en los cuales se almacena el valor en hexadecimal de «expresión», y en formato ya comentado, es decir, primero el menos significativo y detrás el más significativo, como cuando almacenábamos un registro-par en memoria.
Generalmente suelen llevar un nombre simbólico por el cual se va a conocer dicho campo durante todo el programa.

2.5. Directivo DEFM

El formato de este directivo es:

DEFM «expresión»

En general, este directivo sirve para reservar memoria que no esté especificada en los formatos anteriores. Como en ocasiones anteriores ya hemos señalado, también depende de la potencia del traductor el que las expresiones sean más o menos complejas.

3.—Instrucción de No OPerar (NOP)

Esta instrucción no realiza una operación concreta. El microprocesador al encontrarla en un programa lo único que hace es pasar a la instrucción siguiente:
Su formato es:

NOP

y su código máquina, en hexadecimal, correspondiente es # ØØ.
Nos podríamos preguntar, ¿si no hace nada, por qué existe? Pues bien, la respuesta es que *sí* sirve aunque parece que no hace nada.
El microprocesador utiliza un tiempo para ejecutar esta instrucción y eso se utiliza para *perder* tiempo cuando estamos esperando que se produzca algo. También sirve para reservar espacio, a ceros binarios, que posteriormente pueden ser rellenados con información o tal vez, con otras instrucciones.
Aunque sólo fuera por estas dos funciones estaría más que justificada la existencia de esta instrucción dentro del juego del Z-80.



Ganador concurso 48K n.º 1

Fernando Castán Merino

CODIGOS POSTALES

A continuación os ofrecemos el programa ganador del 2.º premio en el 1.er Concurso de Programación de nuestra revista. El afortunado ganador es **Fernando Castán Merino** que nos sorprendió con un programa de códigos postales, permitiéndonos en cualquier momento conocer el prefijo (los dos primeros dígitos) de este número de cinco cifras que compone el código postal de nuestro distrito. Se trata de un programa para aprender más que para jugar, pues demuestra una calidad técnica depurada al menos en lo que a presentación se refiere.

```
   1 GO SUB 1000
   2 POKE 23609,100
   3 REM ***********************
   4 REM *  © FERNANDO CASTAN  *
   5 REM *       PARA 48K      *
   6 REM ***********************
  10 BORDER 5: PAPER 0: CLS : PA
PER 5: INK 2: LET a$="

                          ": GO SUB
20
  12 PAPER 5: INK 6: LET a$="

                                   ":
GO SUB 20
  14 PAPER 5: INK 2: LET a$="

                                   ":
GO SUB 20
  15 GO TO 100
  20 FOR b=1 TO LEN a$: PRINT a$
(b);: BEEP .0019,4: NEXT b: RETU
RN
 100 LET c=1: FOR c=40 TO -40 ST
EP -c: BEEP .01,c: NEXT c
 102 FOR n=-0 TO 60: BORDER INT
(RND*7): BEEP .01,n: BEEP .01,n+
5: NEXT n
 105 BORDER 5: INK 0: FLASH 1: P
RINT AT 21,15;" PULSA UNA TECLA.
": FLASH 0
 106 PAUSE 0
 195 BORDER 5: INK 0: CLS
 197 PRINT : PRINT : PRINT : PRI
NT : LET a$="     CON ESTE PROGRA
MA PODRAS SA-BER EL CODIGO POSTA
L DE TODAS   LAS PROVINCIAS ESPA
NOLAS, PARA  ELLO SOLO TENDRAS Q
UE PULSAR LA INICIAL DE LA PROVI
NCIA QUE DE- SEES SABER SU CODIG
O. EL NUMERO QUE APARECE SON DOS
 CIFRAS DE   LAS CINCO QUE FORMA
N EL CODIGO. PARA TENERLO COMPLE
TO DEBES PO- NER UN CERO Y EL DI
STRITO POSTALQUE TENIA ANTES DE
LA IMPLANTA- CION DE ESTE NUEVO
SISTEMA.": GO SUB 20
 198 FLASH 1: PRINT AT 21,0;" PU
LSA UNA TECLA PARA CONTINUAR.":
FLASH 0: PAUSE 0
 199 CLS
 200 PRINT AT 17,0;"     PULSA LA
 INICIAL DE LA PRO-  VINCIA.
                         PULSA LA
 'w' PARA TERMINAR."
 210 INPUT "                  ";u$
 220 IF u$="a" THEN GO TO 400
 221 IF u$="b" THEN GO TO 405
 222 IF u$="c" THEN GO TO 410
 223 IF u$="g" THEN GO TO 415
 224 IF u$="h" THEN GO TO 420
 225 IF u$="j" THEN GO TO 425
 226 IF u$="l" THEN GO TO 430
 227 IF u$="m" THEN GO TO 435
 228 IF u$="n" THEN GO TO 440
 229 IF u$="o" THEN GO TO 445
 230 IF u$="p" THEN GO TO 450
 231 IF u$="s" THEN GO TO 455
 232 IF u$="t" THEN GO TO 460
 233 IF u$="v" THEN GO TO 465
 234 IF u$="z" THEN GO TO 470
 235 IF u$="w" THEN GO TO 475
 236 IF u$<>"0" THEN GO TO 200
 237 IF u$<>"k" THEN GO TO 200
 400 CLS : PRINT AT 6,6;"ALAVA..
...01                    ALBACET
E..02                    ALICANT
E..03                    ALMERIA
...04                    AVILA..
...05": GO TO 200
 405 CLS : PRINT AT 6,6;"BADAJOZ
...06                    BALEARE
S..07                    BARCELO
NA.08                    BURGOS.
...09": GO TO 200
 410 CLS : PRINT AT 6,6;"CACERES
.....10                  CADIZ..
.....11                  CASTELL
ON...12                  CIUDAD
REAL.13                  CORDOBA
.....14                  CORUNA.
LA...15                  CUENCA.
.....16": GO TO 200
 415 CLS : PRINT AT 6,6;"GERONA.
```

```
.....17                  GRANADA
.....18                  GUADALA
JARA.19                  GUIPUZC
OA...20": GO TO 200
 420 CLS : PRINT AT 6,6;"HUELVA.
..21                     HUESCA.
..22": GO TO 200
 425 CLS : PRINT AT 6,6;"JAEN...
...23": GO TO 200
 430 CLS : PRINT AT 6,6;"LEON...
....24                   LERIDA.
....25                   LOGRONO
....26                   LUGO...
....27": GO TO 200
 435 CLS : PRINT AT 6,6;"MADRID.
...28                    MALAGA.
...29                    MURCIA.
...30": GO TO 200
 440 CLS : PRINT AT 6,6;"NAVARRA
...31": GO TO 200
 445 CLS : PRINT AT 6,6;"ORENSE.
...32                    OVIEDO.
...33": GO TO 200
 450 CLS : PRINT AT 6,6;"PALENCI
A....34                  PALMAS
LAS..35                  PONTEVE
DRA..36": GO TO 200
 455 CLS : PRINT AT 6,6;"SALAMAN
CA.......37              S C de
TENERIFE.38              SANTAND
ER.......39              SEGOVIA
..........40             SEVILLA
..........41             SORIA..
..........42": GO TO 200
 460 CLS : PRINT AT 6,6;"TARRAGO
NA..43                   TERUEL.
```

```
....44                   TOLEDO.
....45": GO TO 200
 465 CLS : PRINT AT 6,6;"VALENCI
A...46                   VALLADO
LID.47                   VIZCAYA
....48": GO TO 200
 470 CLS : PRINT AT 6,6;"ZAMORA.
....49                   ZARAGOZ
A...50": GO TO 200
 475 BORDER 0: PAPER 0: BRIGHT 1
: INK 7: CLS : PRINT AT 12,0;"
MUCHAS GRACIAS POR TU VISITA. ":
 PRINT AT 20,4;"DESCONECTO MIS C
IRCUITOS.": BRIGHT 0: PAUSE 200
 999 CLS : STOP
1000 BORDER 0: PAPER 0: INK 7: C
LS
1001 PRINT : PRINT : PRINT : PRI
NT : PRINT : FOR f=5 TO 15: PRIN
T ;"     ": NEXT f
1002 FLASH 1: PRINT AT 5,5;"FERN
ANDO CASTAN": BEEP .05,0: PRINT
AT 7,7;"     PARA     ": BEEP .
05,2: PRINT AT 9,9;" ---- 48K --
-- ": BEEP .05,4: PRINT AT 11,11
;"   PRESENTA:    ": BEEP .05,6:
PRINT AT 13,13;"   EL CODIGO    "
: BEEP .05,8: PRINT AT 15,15;"
   POSTAL.    ": BEEP .05,10: PAU
SE 50: FLASH 0
1003 INVERSE 1: PRINT AT 21,0;"
PULSA UNA TECLA PARA CONTINUAR."
: INVERSE 0: PAUSE 0
1004 FOR f=1 TO 22: RANDOMIZE US
R 3190: NEXT f: RETURN
```

ANALISIS SOFTWARE

HERBERT'S DUMMY RUN

Como cabía esperar, tras el éxito obtenido por el EVERYONES A WALLY, la prestigiosa firma MICROGEN no desaprovechará la oportunidad de lanzar otro producto dentro de la gama a la que ya nos tiene acostumbrados y que nuevamente ha sido acogido por crítica y público con todos los honores que se merece. Pero ahora, entre nosotros, seamos sinceros..., el programa no es malo, (Dios nos libre de decir nunca nada parecido), tampoco es aburrido, pero... tampoco es nada original y sentimos que así sea. Parece que vivimos la época de las segundas partes, como por ejemplo AVALON y DRAGONTORCH, JET SET WILLY y J.S.W. II (con un previo MANIC MINER), y en este caso no iban a ser menos los de MICROGEN, sólo que encima nadie nos dice que es casi igual a PYJAMARAMA y que tan sólo cambia el personaje, los decorados y ligeramente el argumento. El juego de que hablamos, no es otra cosa que conducir al indómito HERBERT (del Everyones) por los departamentos de unos grandes almacenes, sorteando uno tras otro todos los peligros y completando despacio, muy despacio, el puzzle en que se convierten todos los juegos de la firma. Nunca hemos conseguido completar ningún juego de la serie, pero de lo que sí estamos seguros, es de que en su diseño han colaborado las mentes más complicadas de Inglaterra.



MONSER cada dia +

6 SuperCasettes
Full Memory
por solo 1.795 pts.





Libreto que contiene manual
de instrucciones en Castellano
+Aplicaciones para tu Spectrum













Ya a la venta en Kioscos,
Tiendas Especializadas y
Departamento de
Informatica del El Corte Inglés



MONSER S. A.
C/ Argos, 9 - 28037 Madrid.
Teléfonos: (91) 742 72 12 - 742 72 96